 CMES, vol.15, no.3, pp.147-164, 2006

Remeshing and Refining with Moving Finite Elements. Application to Nonlinear Wave Problems

A. Wacher[1] **and D. Givoli**[2]

Abstract: The recently proposed String Gradient Weighted Moving Finite Element (SGWMFE) method is extended to include remeshing and refining. The method simultaneously determines, at each time step, the solution of the governing partial differential equations and an optimal location of the finite element nodes. It has previously been applied to the nonlinear time-dependent two-dimensional shallow water equations, under the demanding conditions of large Coriolis forces, inducing large mesh and field rotation. Such effects are of major importance in geophysical fluid dynamics applications. Two deficiencies of the original SGWMFE method are (1) possible tangling of the mesh which causes the method's failure, and (2) no mechanism for global refinement when necessary due to the constant number of degrees of freedom. Here the method is extended in order to continue computing solutions when the meshes become too distorted, which happens quickly when the flow is rotationally dominant. Optimal rates of convergence are obtained when remeshing is applied. The method is also extended to include refinement to enable handling of new physical phenomena of a smaller scale which may appear during the solution process. It is shown that the errors in time are kept under control when refinement is necessary. Results of the extended method for some example problems of water hump release are presented.

keyword: moving finite elements, remeshing, global mesh refinement, shallow water equations, Coriolis, wave dispersion, nonlinear waves.

1 Introduction

The class of computational schemes called Moving Finite Element (MFE) methods was introduced for time-dependent problems in [Miller (1981);Miller and Miller (1981)]. There is a wide literature dealing with moving mesh methods, both for finite elements and for finite differences. The text [Baines (1994)] gives an extensive view of MFE methods, and [Budd, Carretero-Gonzalez, and Russell (2005);Li, Liu, and Ma (2004);Tan, Zhang, Huang, and Tang (2004);Tang (2005)] describe recent applications of moving mesh methods. See [Nishioka (2005);Nishioka and Atluri (1980a);Nishioka and Atluri (1980b);Tchouikov, Nishioka, and Fujimoto (2004)] for additional literature on moving finite elements and [Atluri and Zhu (1998);Kim and Atluri (2000)] for moving node meshless methods. The main idea underlying MFE type methods is the optimal distribution of the mesh nodes during the solution process. The determination of the solution $\mathbf{u}$ of the governing partial differential equations and of the optimal node locations are done *simultaneously* in each time step by using a least-squares variational formulation. This process thus differs from that of more familiar adaptive mesh schemes based on *a posteriori* error estimates or error indicators, where the updated mesh is determined *based on* $\mathbf{u}$ rather than simultaneously with it. See, e.g., [Atluri (1992)] for theory and implementation of the more familiar adaptive schemes with applications to aerospace engineering. For a review paper see [Atluri (1984)], which contains a discussion of adaptive mesh methods for linear elasticity and finite-strain problems of inelastic materials. Generally the error-indicator based adaptive mesh schemes involve refinement and sometimes coarsening of the current mesh as the main operations, whereas the original MFE methods involve a geometrical motion of the mesh, with its topology remaining fixed. That is, with given and fixed computational resources (i.e., number of degrees of freedom), MFE methods attempt to find the best distribution of these resources.

The particular MFE method that we concentrate on is the String Gradient Weighted Moving Finite Element (SGWMFE) method, which we will describe below. This method was first developed in 1D in [Wacher, Sobey, and Miller (2003)] and later in detail in [Wacher (2004)]

[1] Faculty of Aerospace Engineering, Technion, Haifa 32000, Israel, Email: abigail@aerodyne.technion.ac.il.

[2] Correspondence to: Faculty of Aerospace Engineering, Technion, Haifa 32000, Israel, Email: givolid@aerodyne.technion.ac.il. Phone: +972(4)829-3814 and Fax: +972(4)829-2030.

for 1D and 2D. The SGWMFE formulation was originally proposed in [Miller (1997)] as an alternative formulation to the Gradient Weighted Moving Finite Element (GWMFE) method, which was developed in detail in [Carlson and Miller (1998a)] and [Carlson and Miller (1998b)] for 1D and 2D systems of partial differential equations. In [Carlson and Miller (1998a)] the authors introduce supporting results that show that GWMFE in 1D efficiently produces accurate results for problems which form steep moving fronts. In [Carlson and Miller (1998b)] the same is extended to 2D, with additional application problems. The results therein show that the method is ideal for problems with sharp moving fronts where one needs to resolve the fine-scale structure of the front to compute the correct answer, greatly improving on the original MFE method found in [Miller and Miller (1981);Miller (1981)]. Recently in [Wacher and Givoli (To appear)] the SGWMFE method is applied to the 2D nonlinear Shallow Water Equations (SWE) including dispersive effects due to Coriolis forces. For other background on moving mesh methods see [Wacher (2004)] where there is an extended reference list on moving mesh methods including MFE, GWMFE and moving finite difference type methods.

The original SGWMFE method has been shown in the references mentioned above to be quite powerful; however it suffers from two main deficiencies. The first one is that during the process in which the finite element nodes are moving, the mesh may eventually get tangled. When this happens the method fails and the simulation must be stopped. This typically happens in vortex dominated flow, or when strongly rotational waves are present, as in the case of geophysical waves under Coriolis force. The second deficiency has to do with the fact that MFE methods take care of *local adaptivity* but neglect *global adaptivity*. Namely, the moving mesh has a fixed number of nodes and elements and a fixed topology. The MFE process aims at distributing the given "resources" in an optimal manner; however, no mechanism for global refinement is incorporated in it. Such global refinement may be necessary to resolve physical phenomena of a smaller scale which are not present initially (hence the initial mesh may ignore them) but appear at a certain time later during the simulation. This may typically happen in geophysical fluid dynamics problems and numerical weather prediction, when "features" such as clouds enter the computational domain.

In this paper we are interested in eliminating these two deficiencies, by combining the MFE approach with refinement and remeshing to handle solutions even when the mesh becomes distorted or the physics becomes more complicated in time. In particular, we show how to extend the SGWMFE method to incorporate these two new capabilities.

The solution of nonlinear time-dependent dispersive wave problems poses a special challenge to the SGWMFE method. They appear, for example, in numerical weather prediction, see [Haltiner (1980);Kalnay, Lord, and McPherson (1998);Steppeler, Hess, Schattler, and Bonaventura (2003)]. Physical wave dispersion gives rise to waves with much richer contents compared to the non-dispersive case. The Coriolis forces tend to both rotate the mesh and extend it radially. In order to continue computations once the mesh becomes too distorted, we consider remeshing. In addition, when new physics is introduced to the equations, for example when new sources appear at a certain time, more nodes may be necessary in order to maintain the same level of accuracy, that is to handle the solutions while the original mesh is not fine enough in view of the new complexity.

In the next section we briefly describe the SGWMFE method for general systems, and discuss the relevant implementation issues. More details can be found in [Wacher (2004);Wacher, Sobey, and Miller (2003)]. In Section 3 we introduce the remeshing algorithm and in Section 4 the refining algorithm. In Section 5 we look at some numerical examples employing the algorithms discussed in this paper, for the solution of the nonlinear dispersive SWE. We also discuss the convergence of the numerical solutions when the remeshing and refining algorithms are applied with SGWMFE. We end the paper with concluding remarks.

2 String Gradient Weighted Moving Finite Elements (SGWMFE)

2.1 Continuous formulation

Consider the system of partial differential equations written in the form:

$$\begin{aligned} u_t &= L_1(u,v,\eta), \\ v_t &= L_2(u,v,\eta), \\ \eta_t &= L_3(u,v,\eta), \end{aligned} \tag{1}$$

for the three unknown functions $\eta(x,y,t)$, $u(x,y,t)$ and $v(x,y,t)$. Here L_1, L_2 and L_3 are given nonlinear spatial differential operators. Specifically we shall consider in this paper the nonlinear dispersive SWE (see (38)). However, the ideas described below are quite general, and no particular reference to the SWE will be made in this section, except in assuming that the problem comprises the three equations (1) with the three unknown functions η, u and v.

The solution graphs for the system (1) may be viewed geometrically and treated as a single evolving two dimensional manifold immersed in five dimensions, that is as $(x,y,u(x,y,t),v(x,y,t),\eta(x,y,t))$. Under reparameterization with moving variables $x(\tau,t),y(\tau,t)$ the manifold becomes an evolving parameterized manifold

$$\mathbf{u}(\tau,t) = (x(\tau,t),y(\tau,t),u(\tau,t),v(\tau,t),\eta(\tau,t))^T, \tag{2}$$

where $\tau = (\tau_1,\tau_2)$ has the meaning of a 'reference location' and geometrically represents the two dimensional surface parametrization of the manifold $[x,y,u,v,\eta]$. The rates of change of the parameterized points in the manifold are then expressed as

$$\dot{\mathbf{u}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{u} \\ \dot{v} \\ \dot{\eta} \end{pmatrix}. \tag{3}$$

Here $\dot{\mathbf{u}}$ and $\mathbf{u}_t$ should not be interpreted to be the same vector, namely $\mathbf{u}_t = (0,0,u_t,v_t,\eta_t)^T$, whereas by the chain rule $\dot{x} = \frac{\partial}{\partial t}x(\tau_1,\tau_2,t)$, $\dot{y} = \frac{\partial}{\partial t}y(\tau_1,\tau_2,t)$, $\dot{u} = u_t + \frac{\partial u}{\partial x}\dot{x} + \frac{\partial u}{\partial y}\dot{y}$, $\dot{v} = v_t + \frac{\partial v}{\partial x}\dot{x} + \frac{\partial v}{\partial y}\dot{y}$ and $\dot{\eta} = \eta_t + \frac{\partial \eta}{\partial x}\dot{x} + \frac{\partial \eta}{\partial y}\dot{y}$. Thus, $\mathbf{u}_t$ can be expressed as a product of a rotation matrix and $\dot{\mathbf{u}}$:

$$\mathbf{u}_t = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -u_x & -u_y & 1 & 0 & 0 \\ -v_x & -v_y & 0 & 1 & 0 \\ -\eta_x & -\eta_y & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{u} \\ \dot{v} \\ \dot{\eta} \end{pmatrix} \equiv \mathbf{R}\dot{\mathbf{u}}. \tag{4}$$

By writing $\mathbf{L}(\mathbf{u}) = (0,0,L_1(u,v,\eta),L_2(u,v,\eta),L_3(u,v,\eta))^T$, we can then define the residual vector $\mathbf{r}$ as

$$\mathbf{r} = \mathbf{R}\dot{\mathbf{u}} - \mathbf{L}(\mathbf{u}). \tag{5}$$

Note that from (1) and (4) $\mathbf{r} \equiv \mathbf{0}$ on the continuous level. On the discrete level, $\mathbf{r}$ will be nonzero but hopefully small. We are now interested in the normal projection of $\mathbf{r}$. We take two linearly independent tangent vectors to the manifold $[x,y,u,v,\eta]$:

$$\mathbf{X} = (1,0,u_x,v_x,\eta_x)^T \qquad \text{and} \qquad \mathbf{Y} = (0,1,u_y,v_y,\eta_y)^T. \tag{6}$$

These two vectors define a plane tangent to the solution manifold in 5D. For any 5D vector, say $\mathbf{r}$, it is possible to construct the normal component $\mathbf{r}_N$ perpendicular to this tangent plane. Simple algebra leads to the result (see [Wacher (2004)]):

$$\begin{aligned} \mathbf{r}_N = \mathbf{r} &- D^{-1}[\| \mathbf{Y} \|^2 (\mathbf{r}\cdot\mathbf{X}) - (\mathbf{Y}\cdot\mathbf{X})(\mathbf{r}\cdot\mathbf{Y})]\mathbf{X} \\ &- D^{-1}[\| \mathbf{X} \|^2 (\mathbf{r}\cdot\mathbf{Y}) - (\mathbf{X}\cdot\mathbf{Y})(\mathbf{r}\cdot\mathbf{X})]\mathbf{Y}. \end{aligned} \tag{7}$$

Here D is the determinant associated with the tangent plane metric, i.e.,

$$D = \begin{vmatrix} \mathbf{X}\cdot\mathbf{X} & \mathbf{Y}\cdot\mathbf{X} \\ \mathbf{X}\cdot\mathbf{Y} & \mathbf{Y}\cdot\mathbf{Y} \end{vmatrix}. \tag{8}$$

Rearranging equation (7), by factoring out $\mathbf{r}$, we can write equation (7) as $\mathbf{r}_N = \mathbf{P}\mathbf{r}$, where the 5×5 projection matrix $\mathbf{P}$ is given by:

$$\begin{aligned} \mathbf{P} = \mathbf{I} &- D^{-1}[\| \mathbf{Y} \|^2 (\mathbf{X}\mathbf{X}^T) - (\mathbf{X}\cdot\mathbf{Y})(\mathbf{X}\mathbf{Y}^T) \\ &+ \| \mathbf{X} \|^2 (\mathbf{Y}\mathbf{Y}^T) - (\mathbf{X}\cdot\mathbf{Y})(\mathbf{Y}\mathbf{X}^T)]. \end{aligned} \tag{9}$$

Here $\mathbf{I}$ is the 5×5 identity matrix. Now, noting that $\mathbf{R} = \mathbf{I} - [\mathbf{X}|\mathbf{Y}|\mathbf{0}|\mathbf{0}|\mathbf{0}]$, it can easily be shown that $\mathbf{P}\mathbf{X} = \mathbf{0}$ and $\mathbf{P}\mathbf{Y} = \mathbf{0}$, which implies that $\mathbf{P}\mathbf{R} = \mathbf{P}$. Thus we can rewrite the equation $\mathbf{r}_N = \mathbf{P}\mathbf{r}$ using equation (5) as

$$\mathbf{r}_N = \mathbf{P}\mathbf{r} = \mathbf{P}(\dot{\mathbf{u}} - \mathbf{L}(\mathbf{u})). \tag{10}$$

The variational formulation of the SGWMFE method consists in finding $\dot{\mathbf{u}}$ so as to minimize the least-squares functional

$$\begin{aligned} \psi[\dot{\mathbf{u}}] &= \int_S \| \mathbf{r}_N \|^2 \, \mathrm{d}S \\ &= \int_S \| \mathbf{P}(\dot{\mathbf{u}} - \mathbf{L}(\mathbf{u})) \|^2 \, \mathrm{d}S \\ &= \int_S (\mathbf{P}(\dot{\mathbf{u}} - \mathbf{L}(\mathbf{u}))) \cdot (\mathbf{P}(\dot{\mathbf{u}} - \mathbf{L}(\mathbf{u}))) \mathrm{d}S. \end{aligned} \tag{11}$$

The differential surface area $\mathrm{d}S$ can be derived by taking the two infinitesimal vectors $\mathbf{X}\mathrm{d}x$ and $\mathbf{Y}\mathrm{d}y$ that span the "parallelogram" of unit area to obtain:

$$\mathrm{d}S = \sqrt{D}\mathrm{d}\Omega = \sqrt{D}\mathrm{d}x\mathrm{d}y, \tag{12}$$

where D is the metric determinant defined in (8).

2.2 Discretization

We consider the finite element approximation of $\mathbf{u}(\tau,t)$ in space. Thus, in each element,

$$\mathbf{u}(\tau,t)=\sum_{j=1}^{Nen} N_j(\tau)\mathbf{u}_j(t), \tag{13}$$

where Nen is the number of element nodes, N_j is the element shape function associated with element node j, and the $\mathbf{u}_j(t)$ are the time-varying nodal values of $\mathbf{u}$. We use a mesh of linear triangular finite elements; thus $Nen=3$, and the finite element approximation is piecewise linear for all of the variables x,y,u,v and η. For simplicity of notation we continue to use the symbols x,y,u,v,η for the finite element approximation of these variables. We note that after the piecewise linear discretization, the derivatives $u_x,v_x,\eta_x,u_y,v_y,\eta_y$, and hence the projection matrix $\mathbf{P}$, are constant in each element. The functional $\psi[\dot{\mathbf{u}}]$ in (11) to be minimized becomes, after discretization, a function of all the nodal rates, $\dot{\mathbf{u}}_j$. To minimize this function, consider its partial derivatives with respect to the five variable rates, and set each to zero to obtain five ordinary differential equations in time at each node. For example we consider the derivative of ψ with respect to the nodal velocities in the x direction, $\dot{x}_I$ (where I is the global node number):

$$\begin{aligned}\frac{1}{2}\frac{\partial\psi}{\partial\dot{x}_I}&=\int_S(\mathbf{P}(\dot{\mathbf{u}}-\mathbf{L}(\mathbf{u})))\cdot\frac{\partial}{\partial\dot{x}_I}(\mathbf{P}(\dot{\mathbf{u}}-\mathbf{L}(\mathbf{u})))\mathrm{d}S\\&=\int_S(\mathbf{P}(\dot{\mathbf{u}}-\mathbf{L}(\mathbf{u})))\cdot(\mathbf{P}\mathbf{E}_1N_I)\mathrm{d}S\\&=\int_S(\mathbf{P}(\dot{\mathbf{u}}-\mathbf{L}(\mathbf{u})))\cdot(\mathbf{E}_1N_I)\mathrm{d}S,\end{aligned} \tag{14}$$

where $\mathbf{E}_1$ is the unit vector $\mathbf{E}_1=(1,0,0,0,0)^T$. The last equality in (14) follows from the fact that $\mathbf{P}$ is an orthogonal projection matrix, i.e. $\mathbf{P}^2=\mathbf{P}$. In an analogous way we take the partial derivatives with respect to $\dot{y}_I$,$\dot{u}_I$,$\dot{v}_I$ and $\dot{\eta}_I$. These are set to zero in order to minimize ψ at each node:

$$\begin{aligned}\frac{1}{2}\left(\frac{\partial\psi}{\partial\dot{x}_I},\frac{\partial\psi}{\partial\dot{y}_I},\frac{\partial\psi}{\partial\dot{u}_I},\frac{\partial\psi}{\partial\dot{v}_I},\frac{\partial\psi}{\partial\dot{\eta}_I}\right)^T&=\int_S(\mathbf{P}(\dot{\mathbf{u}}-\mathbf{L}(\mathbf{u})))N_I\mathrm{d}S\\&=\mathbf{0}.\end{aligned} \tag{15}$$

This finally yields:

$$\int_S\mathbf{P}\dot{\mathbf{u}}N_I\mathrm{d}S=\int_S\mathbf{P}\mathbf{L}(\mathbf{u})N_I\mathrm{d}S. \tag{16}$$

This is a system of nonlinear ordinary differential equations in time. It can be solved using a robust time integrator, such as the Backward Differentiation Formula 2, which is used in this paper, and is outlined in [Carlson and Miller (1998a)]. The integration method used was the one developed and made available by Neil Carlson and Keith Miller described in [Carlson and Miller (1998a);Carlson and Miller (1998b)]. For implementation details and more explicit expressions for the integrals in equation (16) the reader is referred to [Wacher (2004)]. Also note that equation (16) can become very stiff and it is almost always necessary to add small regularization terms to prevent the time integration from failing; see [Carlson and Miller (1998a)] for a discussion on this for GWMFE and [Wacher (2004);Wacher, Sobey, and Miller (2003)] for SGWMFE. For an example of a case where these terms are not necessary see [Wacher (2004)]. For all the experiments in this paper we use a residual error tolerance $=10^{-3}$ and a corresponding viscous regularization term $=5(10)^{-6}$.

2.3 Implementation

2.3.1 Notation and average values

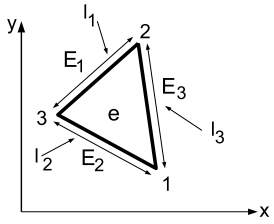

Figure 1 : Triangular element e, its nodes and edges.

The implementation of SGWMFE requires some notation for the average value of a scalar or vector valued function f on an element or an edge. An infinitesimal area on the manifold, $\mathrm{d}S$, is related to the projected $x-y$ area by (12) and since D is constant on each element, the projected area A of an arbitrary element e with area S is related by $S=\sqrt{D}A$. Hence the average of a function f over a triangular element e is given by

$$[f]_{\Omega^e}=\frac{1}{A}\int_{\Omega^e}f\mathrm{d}x\mathrm{d}y=\frac{1}{S}\int_{\Omega^e}f\mathrm{d}S. \tag{17}$$

Here Ω^e is the element domain. We will also assume that the three nodes of the triangular element e are labeled $1,2,3$, as in Fig. 1.

The edge opposite node j is labeled E_j, and its length is denoted ℓ_j. The average of f over an edge E of length ℓ is given by

$$[f]_E = \frac{1}{\ell}\int_E f \mathrm{d}s. \tag{18}$$

where the integrand is with respect to the infinitesimal length along the edge: $\mathrm{d}s = \sqrt{\mathrm{d}x^2 + \mathrm{d}y^2}$.

2.3.2 Calculation of time derivative terms

As $\mathbf{P}$ is constant on an element, the component of the system for time derivatives on a triangular element e, is given by:

$$\int_{\Omega^e} \mathbf{P}\dot{\mathbf{u}} N_j \mathrm{d}S = \mathbf{P}\int_{\Omega^e} \dot{\mathbf{u}} N_j \mathrm{d}S, \tag{19}$$

for $j = 1, \ldots, N_{en}$.

Since the $\dot{\mathbf{u}}$ and N_j are linear functions of the two-dimensional parameter τ, the integrand in equation (19) is quadratic in the components of τ, and hence can be evaluated by the "edge-midpoint rule" which is exact for quadratic functions.

2.3.3 Calculation of flux terms

Consider the case where $L_1 = \nabla\cdot\mathbf{f}$. For the given triangular element e, the contributions from this flux term from the element onto its j node:

$$\begin{aligned}\int_{\Omega^e} \mathbf{P}\begin{pmatrix} 0 \\ 0 \\ -\nabla\cdot\mathbf{f} \\ 0 \\ 0 \end{pmatrix} N_j \mathrm{d}S \\ = \sqrt{D}\mathbf{P}\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}\int_{\Omega^e} (-\nabla\cdot\mathbf{f}) N_j \mathrm{d}x\mathrm{d}y,\end{aligned} \tag{20}$$

for $j = 1, \ldots, N_{en}$. The constancy of $\mathbf{P}$ and D on each element enables these to be taken out of the integral. The scalar integral expression may be integrated by parts as in [Carlson and Miller (1998b)]; e.g., for $j = 1$,

$$\begin{aligned}\int_{\Omega^e} -\nabla\cdot\mathbf{f} N_1 \mathrm{d}x\mathrm{d}y = \int_{\Omega^e} \mathbf{f}\cdot\nabla N_1 \mathrm{d}x\mathrm{d}y \\ - \int_{E_2} N_1 \mathbf{f}\cdot\widehat{\nu}_2 \mathrm{d}s - \int_{E_3} N_1\mathbf{f}\cdot\widehat{\nu}_3 \mathrm{d}s.\end{aligned} \tag{21}$$

Here E_2 and E_3 are edges adjacent to node $j = 1$ and $\widehat{\nu}_2$, $\widehat{\nu}_3$ are outward normals to those edges (N_1 is zero on the third edge). This component can be simplified by using average values. Equation (21) is rewritten in terms of average values as:

$$\begin{aligned}\int_{\Omega^e} -\nabla\cdot\mathbf{f} N_1 \mathrm{d}x\mathrm{d}y = A\nabla N_1\cdot[\mathbf{f}]_{\Omega^e} \\ - \ell_2\widehat{\nu}_2\cdot[N_1\mathbf{f}]_{E_2} - \ell_3\widehat{\nu}_3\cdot[N_1\mathbf{f}]_{E_3}.\end{aligned} \tag{22}$$

2.3.4 Calculation of constant coefficient diffusion terms

Now consider the integration of diffusion (Laplacian) terms. Such terms will appear, for example, when one uses artificial diffusion of the Laplacian type. In this case, a special procedure called *mollification* is used to incorporate these terms in the C^0 least-squares finite element formulation. For piecewise linear approximations, the Laplacian is singular at an edge. In order to overcome this, consider the integrals in neighborhoods of each edge and assume that the piecewise linear approximation is the limit of an approximation which can be differentiated twice; that is the piecewise linear representations of the solution variables need to be mollified so that the first derivatives vary smoothly from element to element in a neighborhood whose thickness will be allowed to vanish. The principle here is the same as in [Carlson and Miller (1998b)], and we follow the notation there.

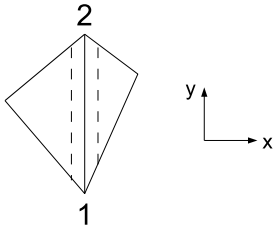

Figure 2 : Mollification of the common edge of two triangles.

Without loss of generality, assume that the 1 to 2 edge E is aligned in the y direction as shown in Fig. 2. This is done so that the unit outward normal vector to E is $\widehat{\nu} = \widehat{x}$, and the unit tangent vector to E is $\widehat{\tau} = \widehat{y}$. Then on crossing the edge, the values of u_y, v_y and η_y have the same constants on both sides of the edge. However, u_x, v_x and η_x have different constant values on the two sides of the edge and experience a jump discontinuity of

difference $2a$, $2b$ and $2c$ respectively, with mean values m_u, m_v and m_η.

Now assume all three are mollified equally, that is the "**X**" tangent vector to the two-dimensional manifold in five dimensions is

$$\mathbf{X}(x) = \mathbf{M} + \sigma(x)\mathbf{A} = \begin{pmatrix} 1 \\ 0 \\ u_x \\ v_x \\ \eta_x \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ m_u \\ m_v \\ m_\eta \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ a \\ b \\ c \end{pmatrix} \sigma(x)\ . \tag{23}$$

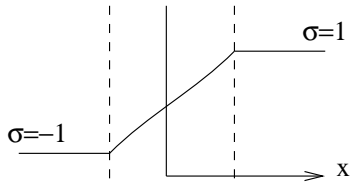

Figure 3 : Mollification function $\sigma(x)$ in the neighborhood of an edge shared by two triangular elements. (The edge shown is perpendicular to the plane of the figure, namely it lies in the x-y plane).

Here $\sigma(x)$ is a function which varies from -1 to 1 in the neighborhood of the edge as shown in Fig. 3. Thus the Laplacian contribution is

$$\mathbf{L} \equiv \begin{pmatrix} 0 \\ 0 \\ u_{xx}+u_{yy} \\ v_{xx}+v_{yy} \\ \eta_{xx}+\eta_{yy} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ a \\ b \\ c \end{pmatrix} \sigma'(x). \tag{24}$$

The "**Y**" tangent vector to the two-dimensional manifold is the constant vector

$$\mathbf{Y} = \begin{pmatrix} 0 \\ 1 \\ u_y \\ v_y \\ \eta_y \end{pmatrix}. \tag{25}$$

Note that the integrals are only considered in a small neighborhood δ of the edge. That is, the neighborhood of the $x-y$ projection of edge E. This is because u_{xx}, v_{xx} and η_{xx} are identically zero away from the edge neighborhood where the rapid variation occurs. First assume the edge E is aligned with the y axis and then rotating the results accordingly. Now take the integral $\int \mathbf{P}\mathbf{L}N_1 \mathrm{d}S$ over the neighborhood δ to get

$$\begin{aligned} &\int_\delta \mathbf{P} \begin{pmatrix} 0 \\ 0 \\ u_{xx} \\ v_{xx} \\ \eta_{xx} \end{pmatrix} N_1(y)\mathrm{d}S \\ &= \int_\delta \sqrt{D}\mathbf{P} \begin{pmatrix} 0 \\ 0 \\ a \\ b \\ c \end{pmatrix} \sigma'(x) N_1(y)\mathrm{d}x\mathrm{d}y \\ &= \int_{y=0}^{y=\ell_{12}} \left(\int_{x_\delta} \sqrt{D}\mathbf{P} \begin{pmatrix} 0 \\ 0 \\ a \\ b \\ c \end{pmatrix} \sigma'(x)\mathrm{d}x \right) N_1(y)\mathrm{d}y \\ &= \frac{\ell_{12}}{2} \int_{\sigma=-1}^{\sigma=1} \sqrt{D(\sigma)}\mathbf{P}(\sigma)\mathbf{A}\mathrm{d}\sigma, \end{aligned} \tag{26}$$

where ℓ_{12} is the length of the edge joining node 1 and node 2, as in Fig. 2. We note that N_1 is equivalent to a function of y alone in an infinitesimal neighborhood of the edge E and is treated as such in equation (26). Further, once the matrix $\mathbf{P}(\sigma)$ has been post-multiplied through by the vector $\mathbf{A}$ (defined in equation (23)) and by $\sqrt{D(\sigma)}$, then

$$\begin{aligned} &\sqrt{D(\sigma)}\mathbf{P}(\sigma)\mathbf{A} \\ &= \frac{1}{\sqrt{D(\sigma)}}[D(\sigma)\mathbf{A} - D_1(\sigma)\mathbf{X}(\sigma) - D_2(\sigma)\mathbf{Y}]. \end{aligned} \tag{27}$$

Here the vector $\mathbf{X}(\sigma)$, and variables $D(\sigma), D_1(\sigma)$ and $D_2(\sigma)$ are functions of σ:

$$\begin{aligned} &\mathbf{X}(\sigma) = \mathbf{M} + \mathbf{A}\sigma, \\ &D(\sigma) = |\mathbf{X}(\sigma)|^2|\mathbf{Y}|^2 - (\mathbf{X}(\sigma)\cdot\mathbf{Y})^2, \\ &D_1(\sigma) = (\mathbf{X}(\sigma)\cdot\mathbf{A})|\mathbf{Y}|^2 - (\mathbf{Y}\cdot\mathbf{A})(\mathbf{Y}\cdot\mathbf{X}(\sigma)), \\ &D_2(\sigma) = (\mathbf{Y}\cdot\mathbf{A})|\mathbf{X}(\sigma)|^2 - (\mathbf{X}\cdot\mathbf{A})(\mathbf{X}(\sigma)\cdot\mathbf{Y}). \end{aligned} \tag{28}$$

Simplifying, the expression $D(\sigma)\mathbf{A} - D_1(\sigma)\mathbf{X}(\sigma) - D_2(\sigma)\mathbf{Y}$ is linear in σ as quadratic terms cancel, and may be written as $\mathbf{G}+\mathbf{H}\sigma$. Here $\mathbf{G}$ and $\mathbf{H}$ are constant vectors in σ. Also notice that $D(\sigma)$ is quadratic in σ and may be written as $c_1+c_2\sigma+c_3\sigma^2$ where c_1, c_2 and c_3 are constants in σ. Using this information we have:

$$\int_{-1}^{1}\sqrt{D(\sigma)}\mathbf{P}(\sigma)\mathbf{A}\mathrm{d}\sigma = \mathbf{G}\int_{-1}^{1}\frac{1}{\sqrt{c_1+c_2\sigma+c_3\sigma^2}}\mathrm{d}\sigma + \mathbf{H}\int_{-1}^{1}\frac{\sigma}{\sqrt{c_1+c_2\sigma+c_3\sigma^2}}\mathrm{d}\sigma. \tag{29}$$

In the one-dimensional theory of SGWMFE [Wacher, Sobey, and Miller (2003);Wacher (2004)] both the integrals multiplied by $\mathbf{G}$ and $\mathbf{H}$ were expressed analytically, since they were also necessary for the calculation of the contributions from the Laplacian terms in one dimension. Due to the sensitivity of these analytic formulas to round-off errors, sixteen point Gauss quadrature is used for the integrals for the studies carried out in the present paper.

Once the entries of the vector have been calculated, it is necessary to rotate the x and y components to the correct orientation, since the calculations were made based on the assumption that the $x-y$ projection of the edge E was aligned with the y axis. This rotation is done by replacing the first two components of the vector (corresponding to its x and y components) and multiplying by the rotation matrix

$$\mathbf{R}_{x,y} = \begin{pmatrix} \frac{\mathrm{d}y}{\ell_{12}} & \frac{\mathrm{d}x}{\ell_{12}} \\ \frac{-\mathrm{d}x}{\ell_{12}} & \frac{\mathrm{d}y}{\ell_{12}} \end{pmatrix}. \tag{30}$$

2.3.5 Calculation of source terms

The contributions over an element from a source term, say $S_1(u,v,\eta)$ in L_1 onto its j^{th} node are obtained as follows:

$$\int_{\Omega^e}\mathbf{P}\begin{pmatrix}0\\0\\S_1(u,v,\eta)\\0\\0\end{pmatrix}N_j\mathrm{d}S = \sqrt{D}\mathbf{P}\begin{pmatrix}0\\0\\1\\0\\0\end{pmatrix}\int_{\Omega^e}S_1(u,v,\eta)N_j\mathrm{d}x\mathrm{d}y, \tag{31}$$

resulting in the equivalent compact form:

$$\int_{\Omega^e}S_1(u,v,\eta)N_j\mathrm{d}x\mathrm{d}y = A[N_jS_1(u,v,\eta)]_{\Omega^e}. \tag{32}$$

3 Remeshing

3.1 The need for remeshing

It has been shown [Wacher (2004);Wacher and Givoli (To appear);Wacher, Sobey, and Miller (2003)] that the main strength of the GWMFE methods is solving problems which contain steep moving fronts. However the main weakness of these methods has been the possibility of mesh tangling. This typically happens for vortex dominated problems, or for strongly rotating wave fields. Mesh tangling causes the method to collapse. One cure to this problem was suggested by A. Kuprat in his PhD thesis, [Kuprat (1992)]. There, he added to the GWMFE method the capability to add and remove nodes as needed when certain mesh geometrical criteria are not met, in order to avoid node tangling. In the two-dimensional case, these criteria are based on tolerances for certain triangle properties, such as the length of an edge and the inscribing radius of a triangle. In [Kuprat (1992)], this method was applied in one and two dimensions to the non-dispersive SWE as well as to a rotating scalar 2D model equation.

In this paper we look at an alternative approach, which is more global in nature. It relies on an observation made in a previous study on the insensitivity of the SGWMFE results on the initial mesh used. In [Wacher (2004)], a study was done on solving the non-dispersive SWE with different initial starting meshes. It is shown there that although at very early times the numerical results may significantly depend on the initial mesh, after a few time steps they "stabilize" and exhibit little sensitivity to it. Moreover, the numerical results, after a short initial time period, converge to solutions with the same level of accuracy regardless of the initial mesh.

In this light, we propose to resolve the problem of node tangling with a scheme which simply remeshes the current solution information onto a new uniform mesh. This is done in the special way described by the algorithm to be outlined in the next subsection. The original SGWMFE method incorporates a check of mesh tangling during the solution process, and if such tangling occurs, the simulation stops. In particular, the method includes a scheme to identify those triangular elements which be-

come too distorted. For each element, a measure of distortion is calculated and compared to a fixed tolerance. For the experiments in this paper this tolerance was taken as 10^{-12}. This tolerance can be set to much larger values to maintain a higher regularity of the mesh; however this would be done at the expense of needing remeshing (or other measures) more frequently. Note that there is a separate check to make sure that the triangular elements always maintain a positive area, as it is necessary to keep the system of equations well defined.

3.2 The remeshing algorithm

The remeshing algorithm may be summarized as follows. Here T denotes a certain discrete time level.

1. $T \longleftarrow 0$, $n \longleftarrow 1$. Fix $\triangle t$ and n_{back}.
2. While ($T \leq T_{final}$)
   (a) For time $T \longleftarrow T$ to $T + n\triangle t$: step in time using SGWMFE.
   (b) If mesh tangling tolerance is crossed:
      i. $n \longleftarrow n - n_{back}$.
      ii. $T \longleftarrow T - n_{back}\triangle t$.
      iii. Interpolate solution at time T onto uniform mesh.
      iv. Go to step 2 (with all solution data corresponding to uniform mesh).
   (c) else: $n \longleftarrow n+1$, go to step 2 (with all solution data unchanged).

When the ordinary SGWMFE solution process stops due to mesh tangling, the motivation for going n_{back} steps back in time to remesh is to allow the solution and new mesh to evolve for a sufficiently long time before reaching the point in time where the older mesh failed. Based on the insensitivity of the numerical results to the initial mesh, as noted above, this procedure is aimed at achieving a smooth transition between the solutions obtained in the old and new meshes. We note that the number of nodes in the new mesh may remain the same as in the old mesh, unless additional global refinement is performed, as discussed in the next section.

3.3 Transfer of information to the new mesh

To transfer information from the old mesh to the new mesh, we use the algorithm proposed in [Zhao, Hobbs, Mühlhaus, and Ord (1999)] for identifying the element in a given mesh containing a given point p. In that paper the algorithm was proposed for quadrilateral elements, and here we have adapted it to triangles. Given a certain triangular element e, we let $\mathbf{a}^j$ be the vector pointing from node j of the element along the edge, counterclockwise, to the subsequent node. Also, we let $\mathbf{b}^j$ be the vector pointing from node j of the element to the given point p. Then the element contains the point p if and only if the following three conditions hold:

$$\begin{aligned} &< \mathbf{a}^1, \mathbf{b}^1 > \ \geq 0, \\ &< \mathbf{a}^2, \mathbf{b}^2 > \ \geq 0, \\ &< \mathbf{a}^3, \mathbf{b}^3 > \ \geq 0. \end{aligned} \tag{33}$$

Here $< \mathbf{a}, \mathbf{b} > := a_1 b_2 - b_1 a_2$.

Using this algorithm, we transfer the data from the old mesh to the new one. For every node p in the new mesh, we first find the element e in the old mesh which contains it. Then we calculate the value of the solution at this point according to the old mesh by using shape function interpolation of the nodal values of element e. By going over the entire set of nodal points of the new mesh in this manner, the entire information is transferred to this mesh.

Numerical experiments that we have performed (not presented here) show that this transfer of information does not hamper the accuracy of the time-varying solution. This is also supported by the rates of convergence obtained (see Section 5). When global refinement is necessary, we also refine the new mesh according to the scheme discussed in the next section.

4 Refining

4.1 The need for refining

The original SGWMFE method incorporates, by definition, *local* refinement of regions in the mesh which require such refinement. However, the method neglects *global* refinement. The moving mesh has a fixed number of nodes and elements and a fixed topology; hence no mechanism for global refinement is incorporated in it. As a result, the density of the initial mesh, which is assumably sufficient for the desired accuracy in the initial phase of the simulation, may not necessarily be fine enough as time proceeds. In particular, it may not be able to resolve physical phenomena of a smaller scale which appear at a certain time later during the simulation. Thus, it is desired to equip SGWMFE with a mechanism for global

refinement, which will ensure that the level of accuracy remains approximately constant throughout the simulation.

Standard Finite Element and Finite Difference methods generally incorporate several kinds of refinement techniques that are used for the handling of new physics. See the Introduction for some pertinent references. The notions of "error estimators" and "error indicators" are defined in this context; these define a measure which is used as a refinement/coarsening criterion. We propose to apply global refinement using a measure of curvature that we call γ. That is, when the measure of curvature γ has grown too much beyond a reference measure, then a new mesh parameter is calculated and the old mesh is replaced with the new finer uniform mesh. The check is not made at every integration time step as this would defeat the purpose of a moving mesh method, but rather after every few "time stations" where a time station typically comprises a few hundreds of time steps.

4.2 The refinement procedure

For each element edge we calculate on each of its two adjacent triangles the gradient of each variable of interest. For simplicity we assign one triangle the label (1) and the second one the label (2). We introduce the following measure γ for the variable ϕ:

$$\gamma_\phi = \frac{1}{N_{iedg}} \sum_{i=1}^{N_{iedg}} \frac{\nabla\phi_i|_{(1)} - \nabla\phi_i|_{(2)}}{\sqrt{\delta x_i^2 + \delta y_i^2}}, \tag{34}$$

where N_{iedg} is the total number of interior edges in the current mesh, and δx_i and δy_i are the $x-$ and $y-$ distances between the centroids of triangles (1) and (2). This curvature measure is reminiscent of the error indicator proposed in [Lohner (1987)] for CFD.

It is also possible, as we do in the examples we consider in this paper, to define γ such that it is determined by the curvature of *two* variables, say ϕ and ψ. A reasonable (non-dimensional) definition is

$$\gamma = \sqrt{\gamma_\phi^2 + \gamma_\psi^2}. \tag{35}$$

At $t = 0$ or at a certain point in time, as the case may be, a reference γ_0 is defined associated with the chosen reference mesh with N_0 nodes. After every few time stations γ is measured and checked against the reference γ_0. If the new γ is larger than a given percentage of the reference γ_0 then the new number of nodes N is calculated using the simple formula

$$N = N_0 \frac{\gamma}{\gamma_0}. \tag{36}$$

Thus the new mesh parameter is

$$h = \sqrt{A(\Omega)}/N\ . \tag{37}$$

A uniform mesh with this h as a mesh parameter is then constructed. After refinement has taken place the reference γ_0 and N_0 are replaced by the new γ and N.

The transfer of information from the old mesh to the new mesh is performed by the same procedure as in the remeshing scheme; see Section 3.3.

4.3 Combined remeshing and refining

The remeshing and refining procedures should be used jointly in the SGWMFE method. To this end, at each time step the mesh quality is checked, and at each "time station" (see previous subsection) the physics complexity γ is evaluated. The mesh is remeshed and/or refined if either of the corresponding criteria is satisfied.

We note that when the need for remeshing arises, it is reasonable to check the value of γ at this instance even if a time station has not been reached yet. Then if the refinement criterion is met, the new mesh is generated with a mesh parameter calculated by (37). This practice has been adopted in the present work.

5 Numerical Examples

5.1 Model problem

5.1.1 Governing equations

The 2D nonlinear SWE including Coriolis force terms are considered. These equations constitute an important basic model in Geophysical Fluid Dynamics, where the Coriolis forces that give rise to dispersive waves originate from the earth's rotation. See, e.g., [Pedlosky (1987);Stoker (1992)].

The SWE in rectilinear coordinates x, y are

$$\begin{aligned} &u_t + \mu u u_x + \mu v u_y - fv = -g\eta_x, \\ &v_t + \mu u v_x + \mu v v_y + fu = -g\eta_y, \\ &\eta_t + \mu u \eta_x + \mu v \eta_y + (h_0 + \mu\eta)(u_x + v_y) = 0. \end{aligned} \tag{38}$$

Here h_0 is the given fluid thickness normal to the $x-y$ plane in the absence of motion, $\eta(x,y,t)$ is the unknown elevation of the fluid above h_0, $u(x,y,t)$ and $v(x,y,t)$ are the fluid velocities in the x and y directions respectively, g is the gravitational acceleration parameter, and f is the dispersion parameter. The binary parameter μ has the value 1 for the nonlinear SWE and 0 for the linearized SWE with zero mean flow; see [Givoli and Neta (2003)]. The first two equations in (38) are the $x-$ and $y-$momentum equations, whereas the third equation is the continuity equation. For a systematic derivation of the SWE see [Pedlosky (1987)] or [Stoker (1992)].

5.1.2 Geometry

We consider the geometrical configuration: $0 \le x \le 10$, $0 \le y \le 10$.

We denote the computational domain Ω.

5.1.3 Initial conditions

We consider a 'hump' of stationary water released at time $t = 0$. In all cases the initial velocities are $u = 0$ and $v = 0$. For the square domain the initial water elevation is

$$\eta(x,y,0) = 0.1[1 + 2e^{-((x-5)^2+(y-5)^2)}]. \tag{39}$$

5.1.4 Boundary conditions

According to the well-posedness theory [Gustafsson, Kreiss, and Oliger (1995)] a single boundary condition is required for (38) at each boundary point. For the square domain we think of the four boundaries as rigid walls, and assign boundary conditions as follows: $v = 0$ on the south and north sides, and $u = 0$ on the west and east sides. These are non-penetrating boundary conditions.

5.1.5 Extra source term

In order to test the refinement algorithm we consider the nonlinear SWE with the introduction of a source term into the x-momentum equation of the form

$$\begin{aligned} &S_1(u,v,\eta) \\ &\quad = e^{-500(t-0.4)^2}[e^{-10((x-5)^2+(y-5)^2)} \\ &\quad + e^{-10((x-2.5)^2+(y-2.5)^2)} + e^{-10((x-7.5)^2+(y-7.5)^2)} \\ &\quad + e^{-10((x-2.5)^2+(y-7.5)^2)} + e^{-10((x-7.5)^2+(y-2.5)^2)}]. \end{aligned} \tag{40}$$

This source term is designed so that it is not felt initially, and as the simulations are approaching a chosen time ($T = 0.4$) the source term becomes strongly active. This is done smoothly in time using a compact Gaussian function of time. In space the source function is defined so that it is a superposition of 5 sources. It should be noted that although the extra source is introduced into the x-momentum equation, it significantly affects all the solution variables due to the coupling among the three equations.

5.1.6 Artificial diffusion

Artificial diffusion is used to maintain continuity of the numerical solution thus avoiding the occurrence of infinite gradients and multivalued foldover. This is done by adding to each of the equations in the system (38) a small artificial diffusion term: $\varepsilon\nabla^2\eta$, $\varepsilon\nabla^2 u$, and $\varepsilon\nabla^2 v$ respectively, with a small coefficient $\varepsilon = 10^{-2}$. Of course, other shock stabilization methods are possible as well; however, the nature of the stabilization employed is not in the focus of the present paper.

For the added artificial diffusion terms we need extra boundary conditions that do not contradict the specified physical boundary conditions. We use zero Neumann on η ($\frac{\partial\eta}{\partial n} = 0$) on all four sides, $\frac{\partial u}{\partial n} = 0$ on the south and north sides, and $\frac{\partial v}{\partial n} = 0$ on the west and east sides.

5.2 Remeshing

We consider the 'water hump release' problem described above. The hump collapses under gravity (for $f > 0$ also the Coriolis forces) for $t > 0$, forming wave fronts which propagate away from the hump's original center and rotate at the same time.

First we examine the convergence of the method with the remeshing algorithm applied at a specified time. We solve the nonlinear SWE ($\mu = 1$ in (38)) in a square domain with $h_0 = 5$ and initial conditions as in equation (39). $g = 9.8$ in (38) for all of the examples presented in this paper. Since an analytic solution is not available, we construct a reference solution (to be regarded as the 'exact solution') by using an extremely fine discretization, much finer than those used for the actual computation. We then define the error measure in the water elevation,

$$E = \| \eta_{ref} - \eta \| / \| \eta_{ref} \|, \tag{41}$$

where η is the actual finite element solution, η_{ref} is the reference solution, and $\| \cdot \|$ is the discrete L_2 norm. Note

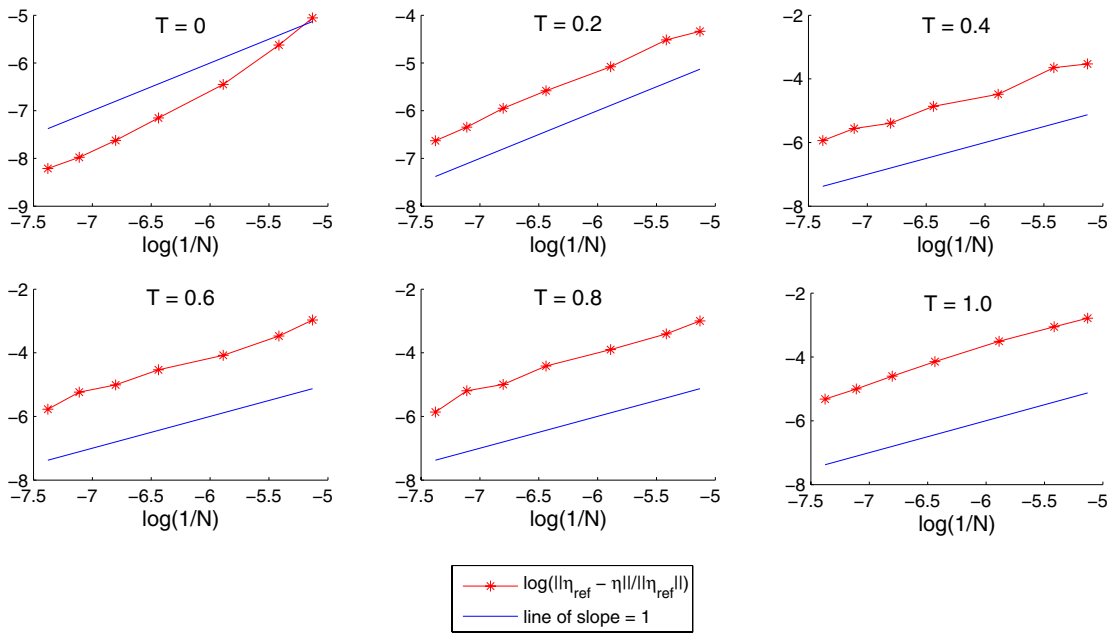

Figure 4 : Convergence in η for the nonlinear SWE, with $h_0 = 5$ and $f = 0$. Remeshing for all values of N took place at $T = 0.4$, except for the reference mesh with $N = 2025$ which was not remeshed.

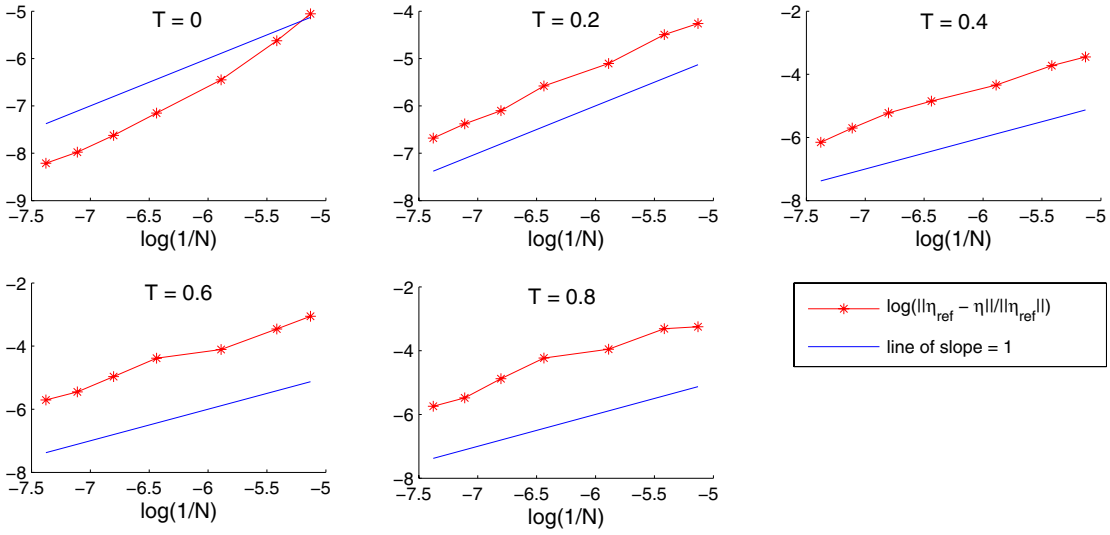

Figure 5 : Convergence in η for the nonlinear SWE, with $h_0 = 5$ and $f = 3$. Remeshing for all values of N took place at $T = 0.4$, except for the reference mesh with $N = 2025$ which was not remeshed.

that the values used for calculating the norms in (41) are interpolated onto a fixed uniform fine mesh in order to enable objective comparison of the various meshes.

The remeshing algorithm has been applied to the non-linear SWE, the non-dispersive case with $f = 0$ as well as two dispersive cases with $f = 3$ and $f = 6$. For the convergence plots the remeshing took place in all cases at the same time station $T = 0.4$ in order for comparison with solutions that were free of remeshing, the comparisons are made with a fine mesh with $N = 2025$ nodes. Fig. 4, Fig. 5 and Fig. 6 show the convergence curves of

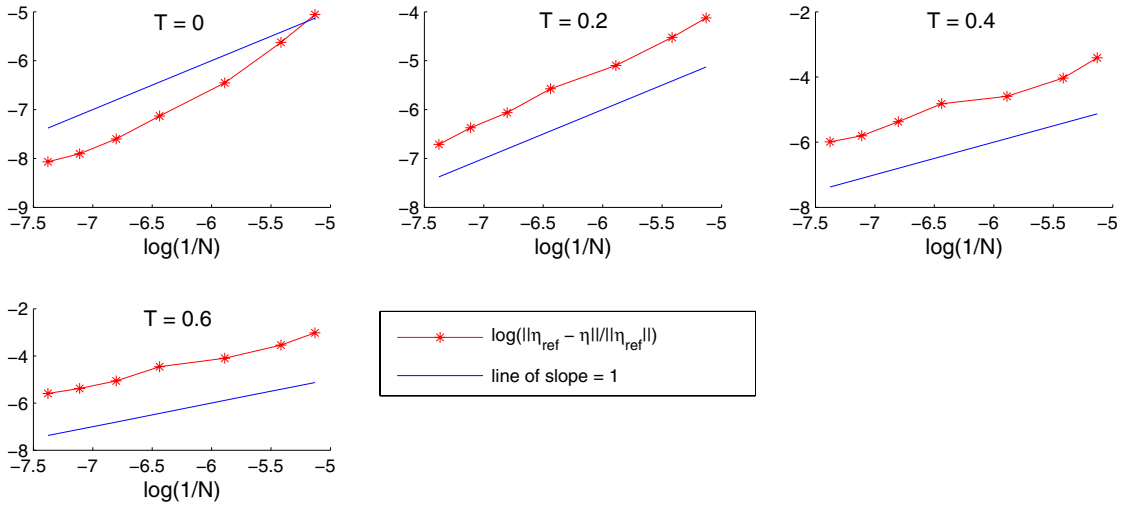

Figure 6 : Convergence in η for the nonlinear SWE, with $h_0 = 5$ and $f = 6$. Remeshing for all values of N took place at $T = 0.4$, except for the reference mesh with $N = 2025$ which was not remeshed.

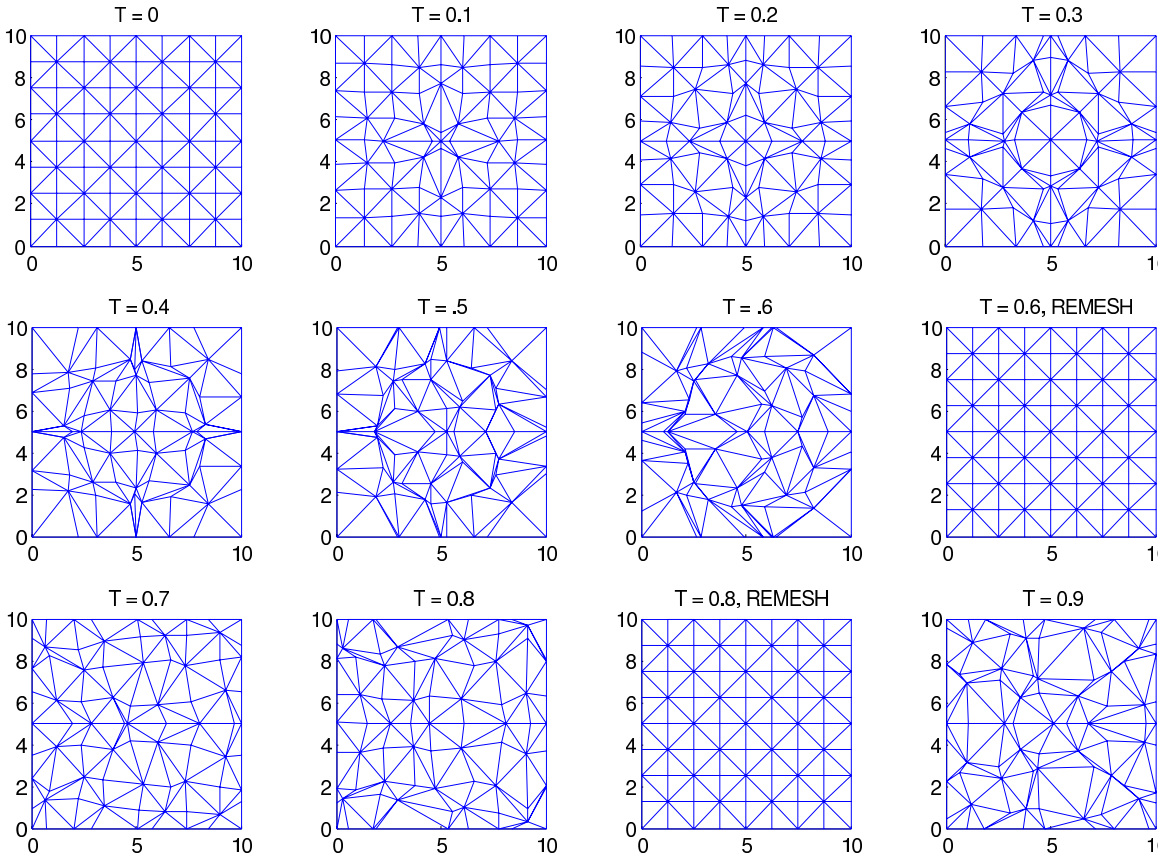

Figure 7 : Mesh for the nonlinear SWE, with $h_0 = 5$, $f = 0$ and $N = 81$.

$E(\eta)$ on a log-log scale, where N is the total number of nodal points. The slopes of the convergence lines for all values of f and t shown, are close to 1. Since the relation between N and the average mesh parameter h is $h = \sqrt{area(\Omega)/N}$, or $N \sim h^{-2}$ this result corresponds to h^2 rate of convergence, which is optimal for linear finite elements. We have performed many more convergence tests, and they all indicated an optimal rate of conver-

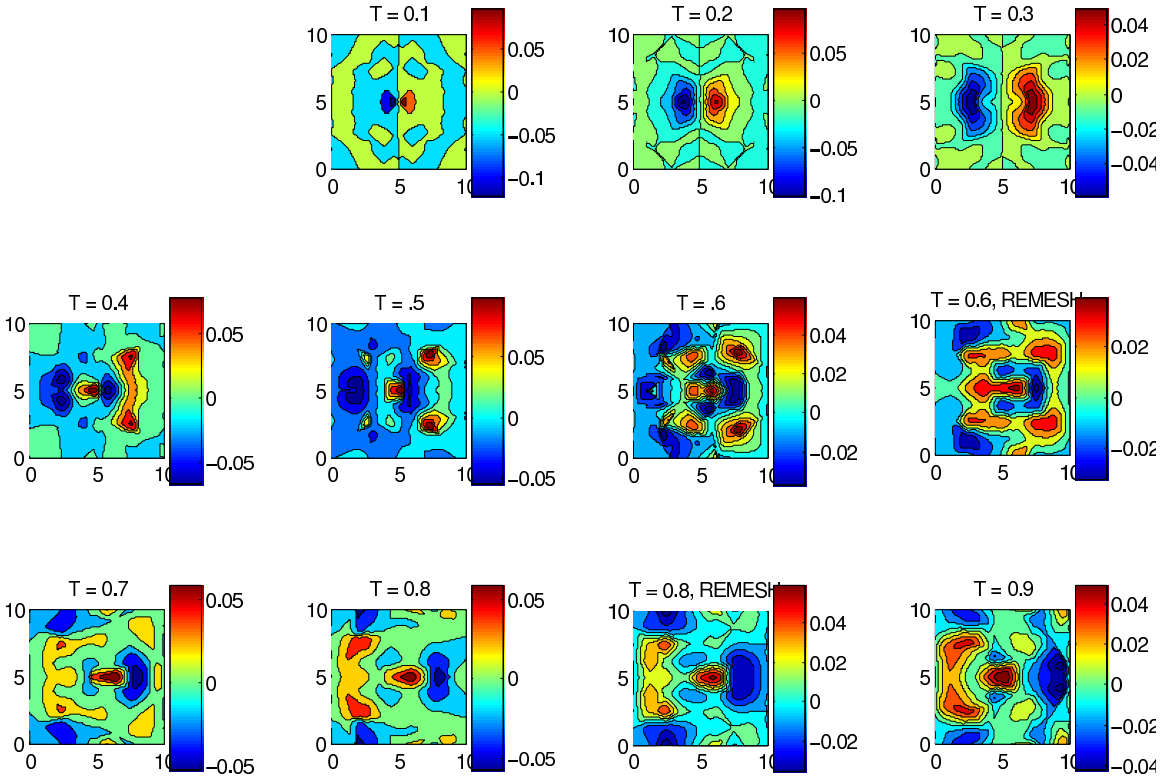

Figure 8 : Solution for the x-velocity u for the nonlinear SWE, with $h_0 = 5$, $f = 0$ and $N = 81$.

gence.

As a second example we consider adding the source term $S_1(u,v,\eta)$ in (40) to the first equation (x-momentum equation) of the SWE. We show results obtained using the remeshing algorithm. Remeshing is applied when the mesh becomes too distorted, see Fig. 7 and 8, for the mesh and solutions of the variable u. For a comparison of the remeshing algorithm when applied to a finer mesh see Fig. 9 and 10.

5.3 Refinement

All of the results in this section are for the nonlinear SWE with $f = 0$, $h_0 = 5$ and include the addition of the source term (40) to the x-momentum equation. In the examples we consider in this paper we consider the γ in equation (35) to have contributions from the two velocity variables in the SWE, we do this by setting $\phi = u$ and $\psi = v$ in equation (35).

Fig. 11 shows the error in the variable η when the refinement and remeshing procedures are both applied, starting with a uniform mesh of $N_0 = 81$. The solutions are compared to the solutions on a fine mesh of $N = 2025$ nodes. The errors of the solutions with the number of nodes of the starting mesh $N = 81$, as well as the refined mesh with $N = 625$ are shown for comparison. Note that the error graph of the coarse mesh (the case $N = 81$) is only shown up until time station $T = 0.9$. This is due to the fact that the method was not able to continue with the computations without refinement beyond this point, unless we perform remeshing.

A plot of the γ for the refinement/remeshing procedure is shown in Fig. 12. The values at the times where the algorithm automatically checks the γ are shown. The values are normalized in order to show the relative values. The percentage of γ_0 used for choosing when to refine the mesh is 30%. See Fig. 13 for mesh plots for the non-dispersive SWE when the refinement and remeshing procedures are applied starting with a uniform mesh of $N_0 = 81$.

Since the source term is added to the x-momentum equation, the solutions of the corresponding variable u are shown for comparison on the initial mesh with 81 nodes, the mesh which has been refined, and the mesh with 625 nodes which is the refined mesh after applying the refinement procedure, see Fig. 8, Fig 14. and Fig. 10 for the corresponding figures.

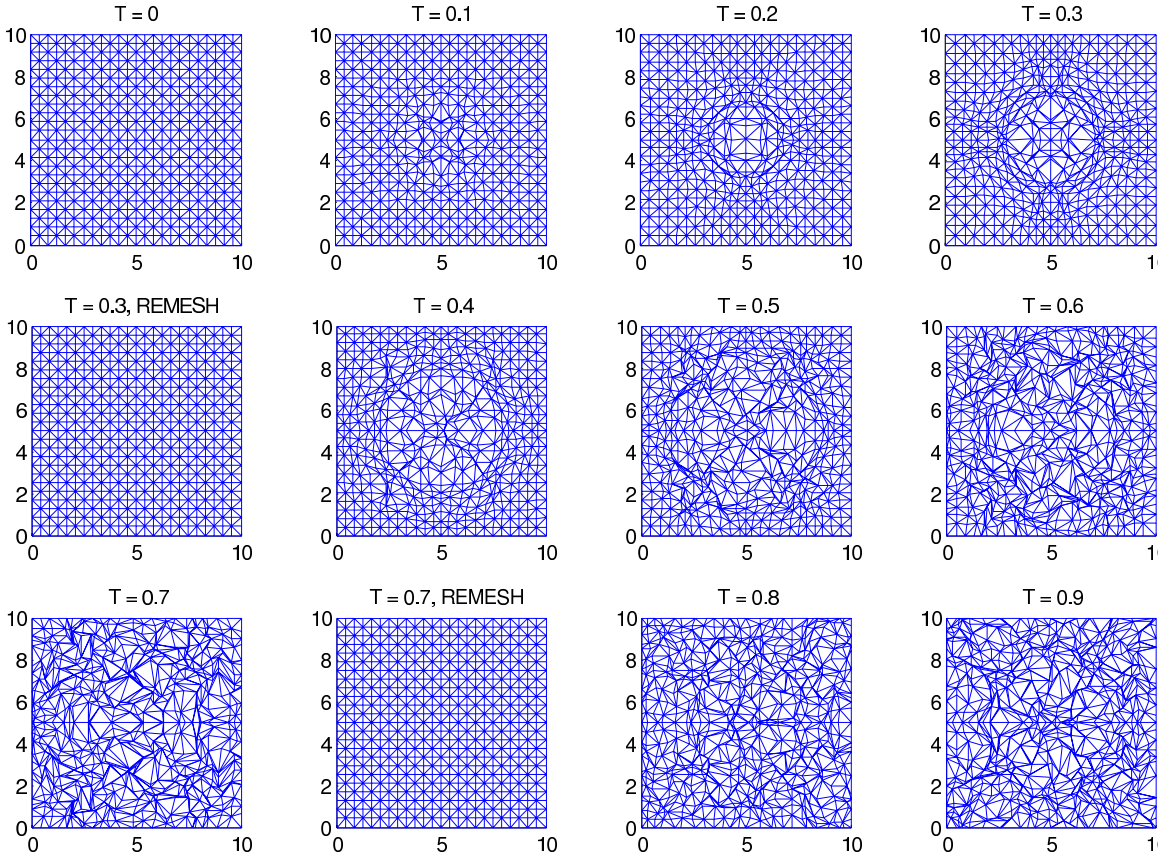

Figure 9 : Mesh for the nonlinear SWE, with $h_0 = 5$, $f = 0$ and $N = 625$.

6 Conclusion

In this paper we proposed two new procedures to be incorporated within the SGWMFE method, namely remeshing and global refinement. These two procedures are introduced in order to overcome the two deficiencies associated with the original SGWMFE, namely (1) the possible tangling of the mesh which may cause the method's failure if not attended, and (2) the lack of a mechanism for global refinement when necessary.

A convergence study of the SGWMFE method with remeshing was performed for the nonlinear SWE in two dimensions for the non-dispersive case as well as two dispersive cases. The results show accurate solutions that correspond to an h^2 rate of convergence, which is optimal for linear finite elements. Further results were presented for the non-dispersive SWE with a source term added to one of the equations thus introducing new physics complexity at some point in time. The refinement algorithm was applied to this problem and the results compared to the solutions with only remeshing using the starting coarse mesh as well as a mesh with the same number of nodes chosen by the refinement algorithm. The results are promising, indicating that the errors in the solutions after refinement are maintained at the same level of accuracy as obtained at the point when refinement became necessary.

In the studies made for this paper, including results not shown here, the remeshing algorithm has been seen to be useful for problems for which the mesh becomes too distorted, allowing one to calculate solutions beyond the point where they otherwise could not. The refining algorithm has shown to be useful when new physics is introduced to a problem and one needs more nodes in order to resolve the new physics.

The implementation of the remeshing and refining algorithms added to the SGWMFE method is straightforward and easily generalized for arbitrary systems of PDEs in multiple dimensions. Together, these two new features eliminate the two main faults of the original SGWMFE method.

Acknowledgement: This work was done while the

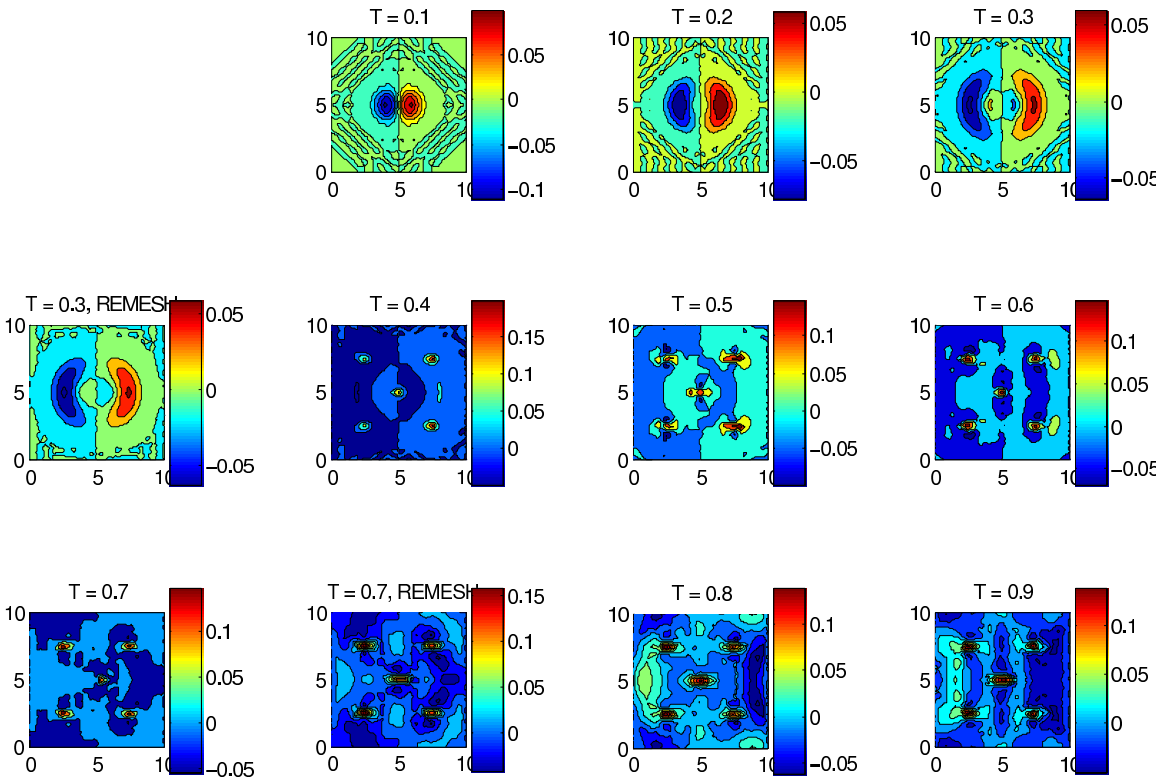

Figure 10 : Solution for the x-velocity u for the nonlinear SWE, with $h_0 = 5$, $f = 0$ and $N = 625$.

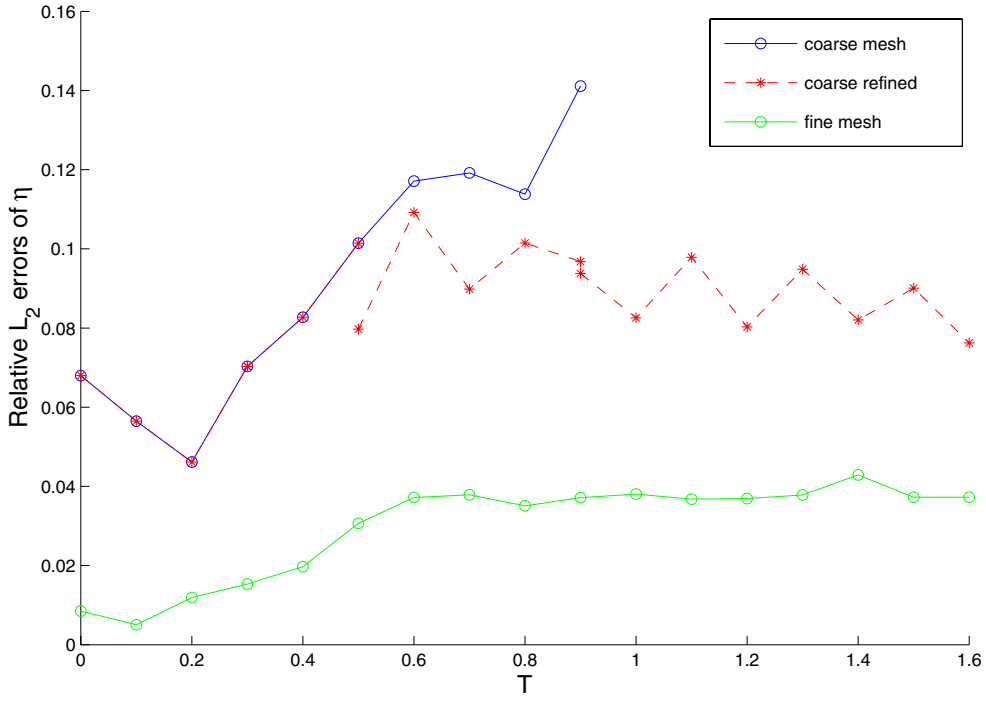

Figure 11 : Error in η for the nonlinear SWE, with $h_0 = 5$, $f = 0$. The coarse mesh uses $N = 81$ nodes, the coarse refined mesh starts with $N_0 = 81$ nodes and refines to $N = 625$ at time station $T = 0.5$, the fine mesh has $N = 625$ nodes. The reference mesh used to compute the error in the solutions has $N = 2025$ nodes.

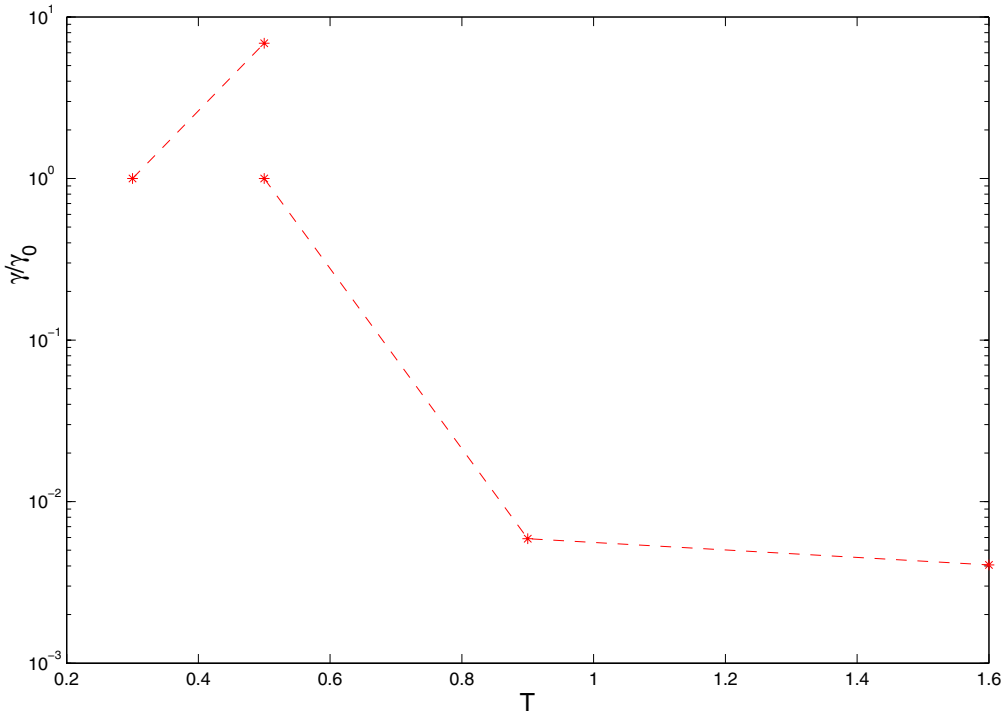

Figure 12 : γ/γ_0 for the nonlinear SWE, with $h_0 = 5$, $f = 0$. The mesh is initially coarse with $N_0 = 81$ nodes and refines to $N = 625$ at time station $T = 0.5$.

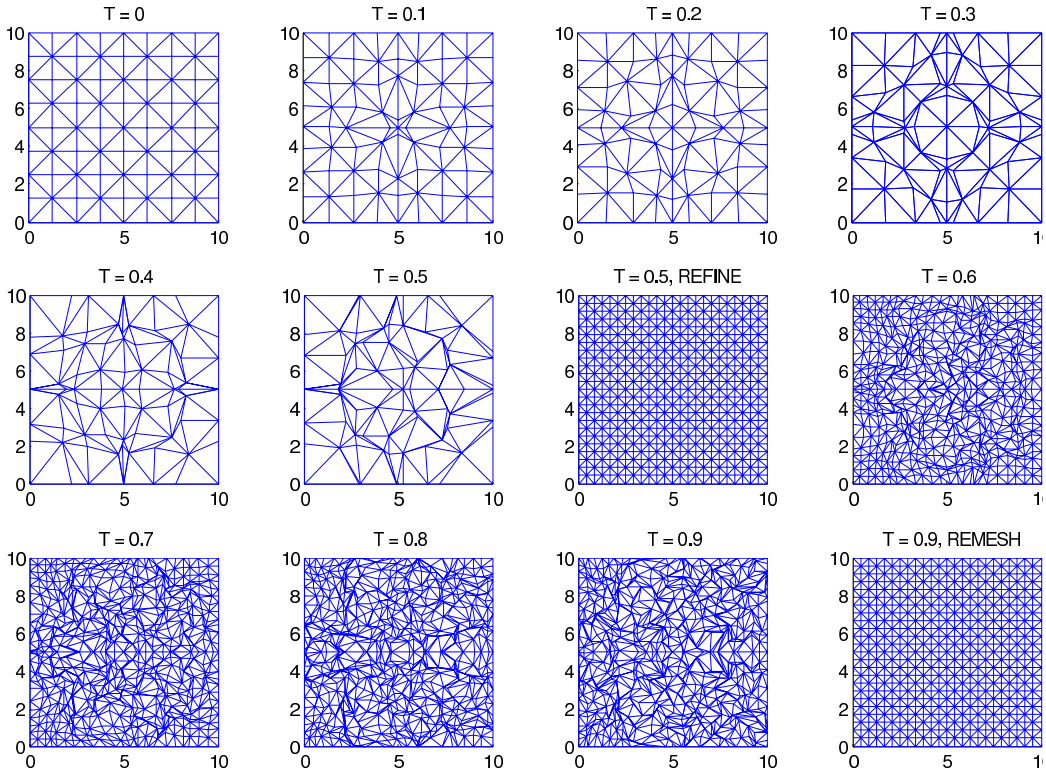

Figure 13 : Mesh for the nonlinear SWE, with $h_0 = 5$, $f = 0$. The mesh is initially coarse with $N_0 = 81$ nodes and refines to $N = 625$ nodes at time station $T = 0.5$.

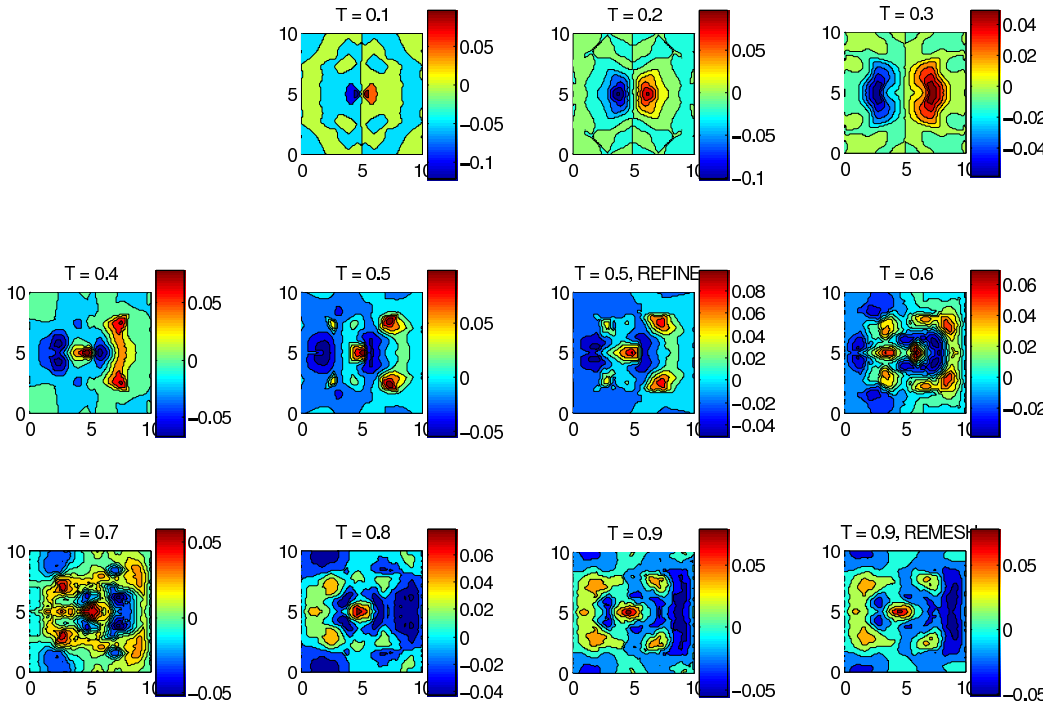

Figure 14 : Solution of x-velocity u for the nonlinear SWE, with $h_0 = 5$, $f = 0$. The mesh is initially coarse with $N_0 = 81$ nodes and refines to $N = 625$ nodes at time station $T = 0.5$.

first author has been supported by the Samuel Neaman Post-doctoral Fellowship. This fellowship is gratefully acknowledged. The second author also acknowledges support from the Fund for the Promotion of Research at the Technion, by the fund provided through the Lawrence and Marie Feldman Chair in Engineering, and by the Seniel Ostrow Research Fund. The authors are grateful to Prof. Josef Stricker for his helpful comments.

References

Atluri, S. N. (1984): Computational Solid Mechanics (Finite Elements and Boundary Elements): -Present Status and Future Directions. *Proceedings of the 4th International Conference on Applied Numerical Modeling*, pp. 29–37.

Atluri, S. N. (1992): *Computational Nonlinear Mechanics in Aerospace Engineering*. Progress in Astronautics and Aeronautics Series, AIAA.

Atluri, S. N.; Zhu, T. (1998): A New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics. *Computational Mechanics*, vol. 22, pp. 117–127.

Baines, M. J. (1994): *Moving Finite Elements*. Oxford University Press Inc, New York.

Budd, C. J.; Carretero-Gonzalez, R.; Russell, R. D. (2005): Precise computations of chemotactic collapse using moving mesh methods. *J. Comput. Phys.*, vol. 202, pp. 463–487.

Carlson, N. N.; Miller, K. (1998): Design and application of a gradient-weighted moving finite element code I: in one dimension. *SIAM Journal on Scientific Computing*, vol. 19, pp. 728–765.

Carlson, N. N.; Miller, K. (1998): Design and application of a gradient-weighted moving finite element code II: in two dimensions. *SIAM Journal on Scientific Computing*, vol. 19, pp. 766–798.

Givoli, D.; Neta, B. (2003): High-order non-reflecting boundary conditions for dispersive waves. *Wave Motion*, vol. 37, pp. 257–271.

Gustafsson, B.; Kreiss, H. O.; Oliger, J. (1995): *Time Dependent Problems and Difference Methods*. Wiley, New York.

Haltiner, G. J. (1980): *Numerical Prediction and Dynamic Meteorology*. Wiley, New York.

Kalnay, E.; Lord, S. J.; McPherson, R. D. (1998): Maturity of operational numerical weather prediction: Medium range. *Bulletin of the American Meteorological Society*, vol. 79, pp. 2753–2769.

Kim, H. G.; Atluri, S. N. (2000): Arbitrary placement of secondary nodes, and error control, in the meshless local Petrov-Galerkin (MLPG) method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 1, no. 3, pp. 11–32.

Kuprat, A. (1992): *Creation And Annihilation Of Nodes For The Moving Finite Element Method, PhD thesis*. University of California, Berkeley.

Li, R.; Liu, W. B.; Ma, H. P. (2004): Moving mesh method with error-estimator-based monitor and its applications to static obstacle problem. *J. Sci. Comput.*, vol. 21, pp. 31–55.

Lohner, R. (1987): An adaptive finite element scheme for transient problems in CFD. *Computer Methods in Applied Mechanics and Engineering*, vol. 61, pp. 323–338.

Miller, K. (1981): Moving Finite Elements II. *SIAM Journal on Numerical Analysis*, vol. 18, pp. 1033–1057.

Miller, K. (1997): A geometrical-mechanical interpretation of gradient-weighted moving finite elements. *SIAM Journal on Numerical Analysis*, vol. 34, pp. 67–90.

Miller, K.; Miller, R. N. (1981): Moving Finite Elements I. *SIAM Journal on Numerical Analysis*, vol. 18, pp. 1019–1032.

Nishioka, T. (2005): Recent Advances in Numerical Simulation Technologies for Various Dynamic Fracture Phenomena. *CMES: Computer Modeling in Engineering & Sciences*, vol. 10, no. 3, pp. 209–216.

Nishioka, T.; Atluri, S. N. (1980): Numerical Modeling of Dynamic Crack Propagation in Finite Bodies, by Moving Singular Elements, Part 1: Formulation. *Journal of Applied Mechanics*.

Nishioka, T.; Atluri, S. N. (1980): Numerical Modeling of Dynamic Crack Propagation in Finite Bodies, by Moving Singular Elements, Part 2: Results. *Journal of Applied Mechanics*, vol. 47, pp. 577–582.

Pedlosky, J. (1987): *Geophysical Fluid Dynamics*. Springer, New York.

Steppeler, J.; Hess, R.; Schattler, U.; Bonaventura, L. (2003): Review of numerical methods for nonhydrostatic weather prediction models. *Meteorology and Atmospheric Physics*, vol. 82, pp. 287–301.

Stoker, J. J. (1992): *Water Waves: The Mathematical Theory with Applications*. Wiley, New York.

Tan, Z.; Zhang, Z.; Huang, Y.; Tang, T. (2004): Moving mesh methods with locally varying time steps. *J. Comput. Phys.*, vol. 200, pp. 347–367.

Tang, T. (2005): Moving mesh methods for computational fluid dynamics. Technical Report CSCAMM-05-04, University of Maryland, Centre for Scientific Computation and Mathematical Modeling Report, 2005.

Tchouikov, S.; Nishioka, T.; Fujimoto, T. (2004): Numerical Prediction of Dynamically Propagating and Branching Cracks Using Moving Finite Element Method. *CMC: Computers, Materials, and Continua*, vol. 1, no. 2, pp. 191–204.

Wacher, A. (2004): *String Gradient Weighted Moving Finite Elements for Systems of Partial Differential Equations, DPhil thesis*. Computing Laboratory, Oxford University.

Wacher, A.; Givoli, D. (To appear): Solution of nonlinear dispersive wave problems using a moving finite element method. *Communications in Numerical Methods in Engineering*.

Wacher, A.; Sobey, I.; Miller, K. (2003): String gradient weighted moving finite elements for systems of partial differential euqations. Technical Report 03/15, Computing Laboratory, Oxford University, Numerical Analysis Group, 2003.

Zhao, C.; Hobbs, B. E.; Mühlhaus, H. B.; Ord, A. (1999): A Consistent Point-Searching Algorithm for Solution Interpolation in Unstructured Meshes Consisting of 4-Node Bilinear Quadrilateral Elements. *Int. J. Numer. Meth. Engrng.*, vol. 45, pp. 1509–1526.